IOT

Simulation-Based Performance Evaluation of Temporal Filtering for Angle-Scanned Ultrasonic Intrusion Monitoring

Chagalamarri Rajanikanth[1], Ediga Varunsai[1], Agaventi Santhosh[1], Shaik Riazun[2]

[1]Student Researcher, Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies (RGUKT), RK Valley, Andhra Pradesh, India
[2]Assistant Professor (Project Guide), Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies (RGUKT), RK Valley, Andhra Pradesh, India

Published November 2026

Abstract

Background/Objective: Short-range ultrasonic ranging is widely used in low-cost obstacle and perimeter monitoring, but single-sample threshold decisions frequently trigger false alarms under acoustic noise and transient outliers. This study evaluates deterministic temporal filtering strategies for an angle-scanned ultrasonic intrusion-monitoring system. **Methods:** Three lightweight decision rules suitable for 8-bit microcontrollers are compared: (A) direct distance thresholding at 20 cm, (B) a causal three-sample median filter followed by thresholding, and (C) causal median filtering followed by a two-consecutive-positive temporal confirmation rule. Algorithmic robustness is quantified across 30 independent Python simulation trials per Gaussian noise level (σ = 0.5 to 6.0 cm) with injected ±25 cm impulse outliers, alongside target-distance sweeps and 200-trial response-latency tests, complemented by a Tinkercad Arduino Uno virtual circuit model. **Results:** Causal median filtering reduces sample-level false-alarm rates from 1.45–2.07% (baseline) to 0.31–0.37% across all tested noise levels, and improves mean detection rate at σ = 6.0 cm from 79.32% to 87.17%. Adding two-sample temporal confirmation lowers sample-level false alarms further to 0.26–0.32% and pre-event trial false alarms from 83.0% to 4.5% at σ = 6.0 cm, while increasing mean response delay to 2.20–2.66 samples. **Conclusion:** The results quantify a clear robustness-latency trade-off for embedded ultrasonic sensing and provide a structured baseline for physical hardware validation.

Keywords: Arduino Uno; false alarm reduction; intrusion monitoring; median filter; temporal confirmation; ultrasonic sensing

1 Introduction

Ultrasonic distance sensors estimate range from the travel time of an emitted acoustic pulse and its returning echo. Their low cost and simple digital interfaces make them attractive for educational prototypes, short-range obstacle detection, and low-complexity monitoring systems. When mounted on a rotating servo, a sensor can sample multiple angular directions and provide a rudimentary scan of the surrounding area. Such a system can indicate a nearby object through a display and audible or visual outputs.

A practical challenge is that a single distance measurement is not a perfect representation of the environment. Echo strength, target orientation, surface material, nearby objects, and electronic or acoustic interference can cause unstable readings. A direct threshold decision is simple and fast, but

any noisy sample that crosses the threshold may produce an alarm. Filtering and temporal confirmation can suppress isolated crossings, although they may delay the alarm or miss short events.

This paper compares three lightweight decision strategies using Python simulation results. The aim is not to claim a validated security product, but to quantify how filtering changes detection rate, false-alarm behavior, and response delay under controlled synthetic noise. The circuit simulation and algorithm evaluation are treated as distinct evidence sources.

The motivation for this comparison is practical: an alarm that triggers on every brief threshold crossing may be fast but difficult to use, whereas a highly persistent decision can suppress nuisance alerts at the cost of reacting later. A useful design therefore requires more than a single accuracy score. This study evaluates three transparent decision rules using identical synthetic measurement assumptions, identifying trends within the model and highlighting which claims still require testing on physical hardware.

1.1 Related work and design context

Ultrasonic sensors have been widely used in radar-like obstacle-detection demonstrations. Suaif and Gani [1] describe an implementation of an ultrasonic sensor for obstacle detection, illustrating the relevance of range sensing in low-cost embedded systems. Work on ultrasonic signal filtration also highlights the importance of processing measured signals before making obstacle decisions; Sachkov and Yusupova [2] discuss filtration algorithms for ultrasonic sensors used in obstacle-detection devices. Prior studies on autonomous land vehicle obstacle detection [3], outdoor blind navigation [4], mobile robot sonar avoidance [5,6], automotive ultrasonic ranging [7], robotic distance measurement [8], and Arduino-based ultrasonic radar prototypes [9] further motivate lightweight deterministic filtering rather than complex classifiers.

The present study evaluates three methods with increasing decision persistence. Method A is a baseline comparator representing the minimum-processing approach. Method B uses a causal median of three recent measurements, a non-linear robust statistic [10] that rejects a single isolated extreme value when the other two samples are consistent. Method C adds a temporal confirmation requirement: after median filtering, two consecutive positive decisions are required before an intrusion is confirmed. All three methods are intentionally compact enough to run deterministically on an 8-bit microcontroller.

1.2 System concept and research questions

The virtual prototype consists of an Arduino Uno, ultrasonic distance sensor, servo motor, 16×2 I2C LCD, red and green LEDs, and a piezo buzzer. The servo provides angular positioning for a scan from 0° to 180°. The conceptual controller reads distance, compares the measurement with a nominal 20 cm threshold, and updates the LCD status while driving the LEDs and buzzer as a local alert interface. Component behavior follows published board and sensor documentation [11,12]. This represents a virtual circuit concept rather than measured physical hardware performance.

The evaluation addresses three primary research questions: (1) does median filtering improve detection and reduce false alarms as measurement noise increases; (2) what additional effect results from requiring two consecutive positive decisions; and (3) how do target distance and decision persistence affect the balance between missed detections, false alarms, and response delay? The answers are bounded by the model assumptions and parameter ranges used in the simulation.

A median filter is attractive in this setting because it is non-parametric and has a fixed, small computational cost. For a three-sample window, it selects the middle observation rather than the arithmetic average, preventing a single extreme outlier from dominating the filtered output. However, if a disturbance persists across most of the window, the median can also follow that disturbance. Temporal confirmation operates on the binary decision rather than raw range values, suppressing isolated single-sample triggers at the cost of requiring persistent events.

2 Methods

2.1 Synthetic noise simulation model

The algorithmic evaluation was performed in Python using a synthetic distance-measurement model. The scenario represented a target at 15 cm and an empty-scene distance of 35 cm for the main noise comparison, with a nominal intrusion threshold of 20 cm. Gaussian measurement noise was tested at standard deviations $\sigma \in \{0.5, 1, 2, 4, 6\}$ cm. The model also included an impulse-outlier component with probability 0.03 and magnitude of ±25 cm, with values clipped to the model's stated measurement range of 0–400 cm.

For the main comparison, 30 independent trials were run for each noise condition, reporting means and standard deviations across trials. A separate distance experiment varied the target distance across 10, 15, 18, 20, and 25 cm. A response-time experiment used sequences containing 100 pre-event, 100 event, and 100 post-event samples, with 200 trials per noise condition. Because the simulation reports delay in sample counts, delay values are analyzed directly in sample units.

2.2 Decision strategies and evaluation metrics

- **Method A (Direct Threshold):** A raw measurement below the nominal 20 cm threshold is treated immediately as a positive intrusion indication. This method exhibits minimal algorithmic delay but is sensitive to transient downward noise.
- **Method B (Causal Median):** The current measurement is combined with the two preceding measurements, and the median of the three is compared against the threshold. The filter is strictly causal and relies only on current and past samples.
- **Method C (Median Plus Confirmation):** The median-filtered decision must remain positive for two consecutive samples before the intrusion event is confirmed, trading additional latency for higher immunity to brief crossings.

Detection rate is defined as the proportion of simulated target-present opportunities correctly detected. Sample-level false-alarm rate measures the proportion of evaluated non-event samples incorrectly classified as an intrusion. Response delay is the number of simulation samples elapsed between event onset and the first confirmed detection. The response-time experiment additionally reports the percentage of trials exhibiting any false alarm prior to event onset. Keeping the nominal threshold fixed at 20 cm across all three rules isolates the effect of temporal decision logic.

2.3 Two-stage embedded decision pipeline

To bridge low-cost microcontroller sensing with reliable perimeter monitoring, we propose a deterministic, two-stage temporal filtering and confirmation architecture for an angle-scanned ultrasonic intrusion-detection system. Rather than relying on computationally intensive statistical estimators or multi-sensor fusion arrays, the proposed architecture is tailored for resource-constrained 8-bit

microcontrollers (such as the ATmega328P) where deterministic timing and low memory overhead are essential.

The proposed system operates across four sequential functional blocks:

1. **Angular Sweep and Echo Acquisition:** A servo motor steps an HC-SR04 ultrasonic transceiver across a 0° to 180° sector at fixed 60 ms intervals. At each angular step, three rapid ultrasonic range pings separated by 5 ms are triggered, converting echo pulse width t_{echo} into range $d = (v_s \cdot t_{echo})/2$ using the nominal ambient acoustic velocity v_s = 343 m/s. Timed-out echoes are mapped to a ceiling sentinel value (400 cm) to prevent zero-value false triggers.
2. **Stage-1 Non-Linear Outlier Rejection (Causal Median):** The three acquired range samples {d[n], d[n-1], d[n-2]} are sorted in ascending order, and the median $d_{med}[n]$ = median(d[n], d[n-1], d[n-2]) is extracted. This non-linear stage suppresses single-sample acoustic multipath spikes, electrical glitches, and isolated impulse outliers (±25 cm) without smearing sharp range transitions at object boundaries.
3. **Stage-2 Perimeter Thresholding and Temporal Confirmation:** The filtered distance $d_{med}[n]$ is compared against a nominal perimeter boundary $T_{intrusion}$ = 20 cm to produce an intermediate binary state b[n] = I($d_{med}[n] \leq 20$ cm). A persistence counter c[n] increments when b[n] = 1 and resets immediately to 0 when b[n] = 0. An intrusion state A[n] = 1 is asserted only when c[n] ≥ 2 (two consecutive positive median-filtered evaluations).
4. **Multi-Modal Local Alert and Telemetry Interface:** Upon alarm assertion (A[n] = 1), the controller activates a red visual indicator, deactivates the green safe-status LED, drives a 1 kHz audible buzzer, and renders the instantaneous intrusion angle and range on a 16×2 I2C LCD while streaming diagnostic telemetry at 9600 baud.

On an 8-bit microcontroller without hardware floating-point units, maintaining a three-sample sliding range buffer requires only 6 bytes of SRAM per active scan sector, while sorting three integers requires at most three conditional compare-and-swap operations. The binary confirmation stage adds a single 1-byte state counter. Within each 60 ms angular dwell window, three consecutive range measurements spaced by 5 ms require 15 ms of active acoustic sampling, leaving 45 ms for median sorting, threshold confirmation, I2C display refresh, and servo settling. To prevent spurious startup triggers, the sliding buffer is initialized to the 400 cm sentinel value and c[n] is held at zero until three valid sample cycles have elapsed.

2.4 Tinkercad virtual circuit implementation

This section records the virtual-circuit implementation supplied with the project to complement the Python noise experiments. The Arduino sketch controls an HC-SR04 ultrasonic transceiver mounted on a micro-servo scanning from 0° to 180°, an I2C 16×2 LCD initialized at address 0x27, red and green status LEDs, and a 1 kHz piezo buzzer, logging angle, distance, and status over serial at 9600 baud, with pin assignments and circuit states detailed in Table 1, Figure 1, and Figure 2.

Table 1. Tinkercad virtual prototype hardware pinout and operating parameters.

Component / Signal	Arduino Connection / Setting
Ultrasonic TRIG / ECHO	Digital Pins D9 / D10
Green LED / Red LED	Digital Pins D2 / D3
Piezo Buzzer / Servo Signal	Digital Pins D4 / D6

Component / Signal	Arduino Connection / Setting
16×2 LCD Display	I2C Interface (Address 0x27)
Intrusion Threshold	20 cm
Confirmation Rule	2 consecutive near readings
Servo Scan Interval	60 ms (3 pings at 5 ms spacing)

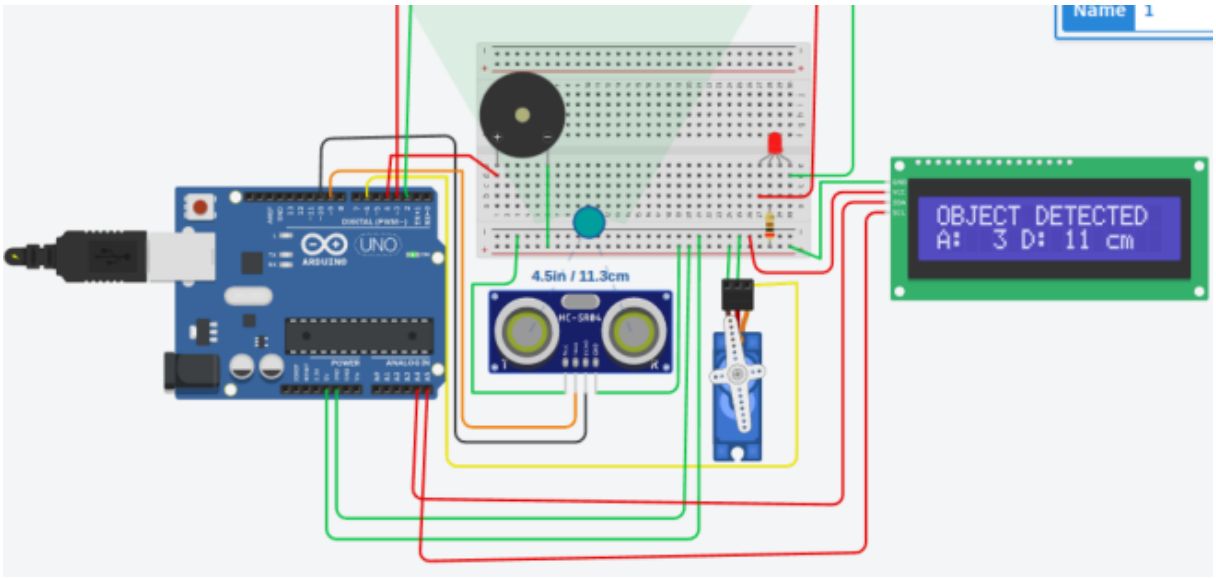


Figure 1. Tinkercad virtual circuit in the object-detected state, showing the Arduino Uno, HC-SR04 sensor (11.3 cm), micro-servo, red LED, piezo buzzer and LCD displaying OBJECT DETECTED (A: 3, D: 11 cm).

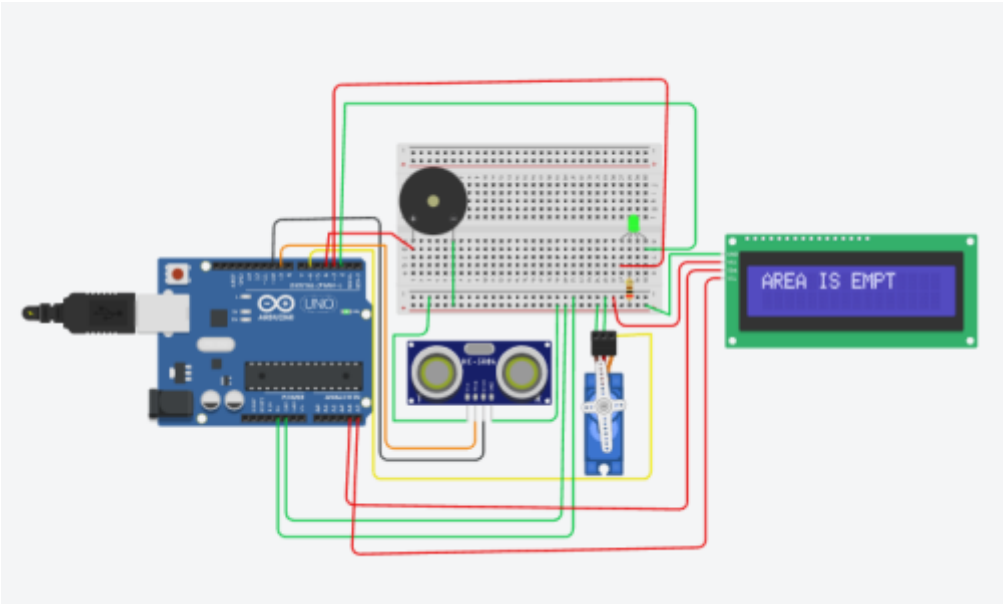


Figure 2. Tinkercad virtual circuit in the quiescent state, showing the Arduino Uno, sensor, servo, buzzer, green LED and I2C LCD wiring layout displaying AREA IS EMPTY.

In the Tinkercad firmware logic, each filtered distance is computed from three readings separated by 5 ms. If all three readings time out, the program reports no echo; otherwise, invalid readings are replaced with 400 cm before sorting to extract the median. When the median distance is at most 20 cm, a near-reading counter increments up to two and resets on any non-near reading. Once the

counter reaches two, the system asserts the object-detected state, illuminates the red LED, extinguishes the green LED, and enables a 1 kHz buzzer tone.

3 Results

3.1 Noise robustness across 30 independent trials

Table 2 and Figure 3 summarize the main 30-trial noise experiment. At low and moderate noise, all three methods maintain high detection rates. As σ rises, the direct threshold baseline degrades more quickly. At σ = 6 cm, the median method reaches 87.17% mean detection, compared with 79.32% for direct thresholding. The confirmation method yields 80.84% at the same noise level, illustrating that stronger persistence does not necessarily improve detection when noise and threshold crossings become more challenging.

Table 2. Mean detection (Det.) and sample-level false-alarm (FA) rates across 30 trials.

σ (cm)	Method A Det. (%)	Method B Det. (%)	Method C Det. (%)	Method A FA (%)	Method B FA (%)	Method C FA (%)
0.5	98.76	98.18	96.43	1.59	0.37	0.32
1.0	98.74	98.17	96.43	1.45	0.36	0.31
2.0	97.74	98.10	96.31	1.55	0.35	0.30
4.0	88.43	94.09	90.23	1.57	0.31	0.26
6.0	79.32	87.17	80.84	2.07	0.34	0.26

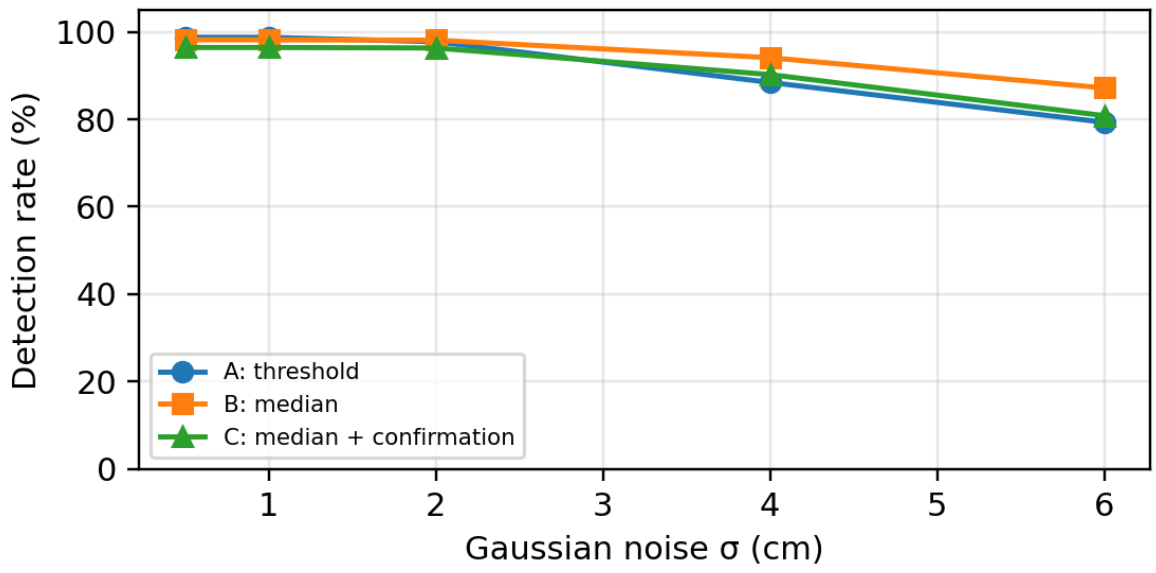


Figure 3. Detection rate versus Gaussian noise standard deviation σ for Methods A, B and C.

The false-alarm comparison shown in Figure 4 demonstrates a consistent advantage for temporal processing. Across the tested noise levels, the baseline false-alarm mean ranges from 1.45% to 2.07%. The median method reduces this range to 0.31–0.37%, whereas the median-plus-confirmation method achieves 0.26–0.32%. Thus, the median filter provides a substantial reduction in sample-

level false alarms relative to direct thresholding, while confirmation provides an incremental further reduction.

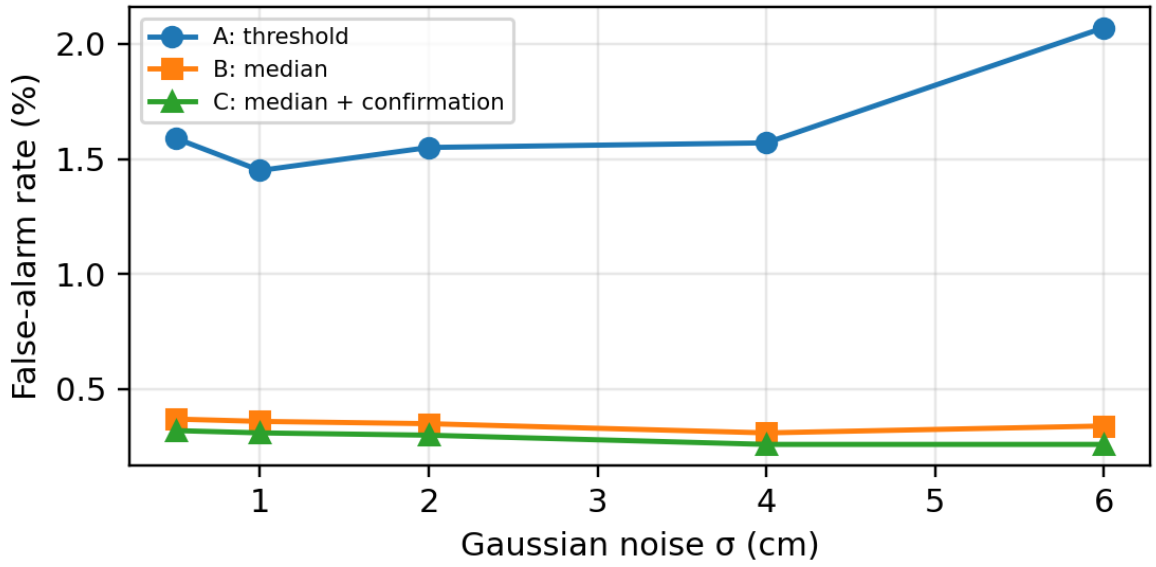


Figure 4. Sample-level false-alarm rate versus Gaussian noise standard deviation σ.

The reported standard deviations generally rise at larger noise levels, particularly for detection rate under the confirmation rule, reflecting sensitivity to specific noise sequences near the 20 cm threshold. At σ = 4 cm, median filtering raises mean detection from 88.43% to 94.09% while lowering sample-level false alarms from 1.57% to 0.31%. At σ = 0.5–2 cm, baseline detection is already near 98%, so filtering primarily delivers operational value by suppressing nuisance false alarms. The confirmation method produces the lowest false-alarm means in Table 2 but trades some sensitivity for additional alarm stability.

3.2 Target distance sensitivity and response latency

Target distance materially changes the classification problem. At σ = 4 cm, detection is high for a target at 10 cm (well inside the perimeter) but declines as the target approaches and crosses the nominal 20 cm threshold. At 25 cm, the target lies outside the nominal perimeter and detections are rare, consistent with the intended threshold logic. Table 3 and Figure 5 summarize selected results from the distance experiment.

Table 3. Selected detection rates across target distances at σ = 4 cm.

Target Distance	Method A Det. (%)	Method B Det. (%)	Method C Det. (%)
10 cm	97.94	99.40	98.86
15 cm	88.35	95.62	93.05
18 cm	68.47	75.63	66.26
20 cm	50.03	49.48	37.02
25 cm	12.37	3.91	1.79

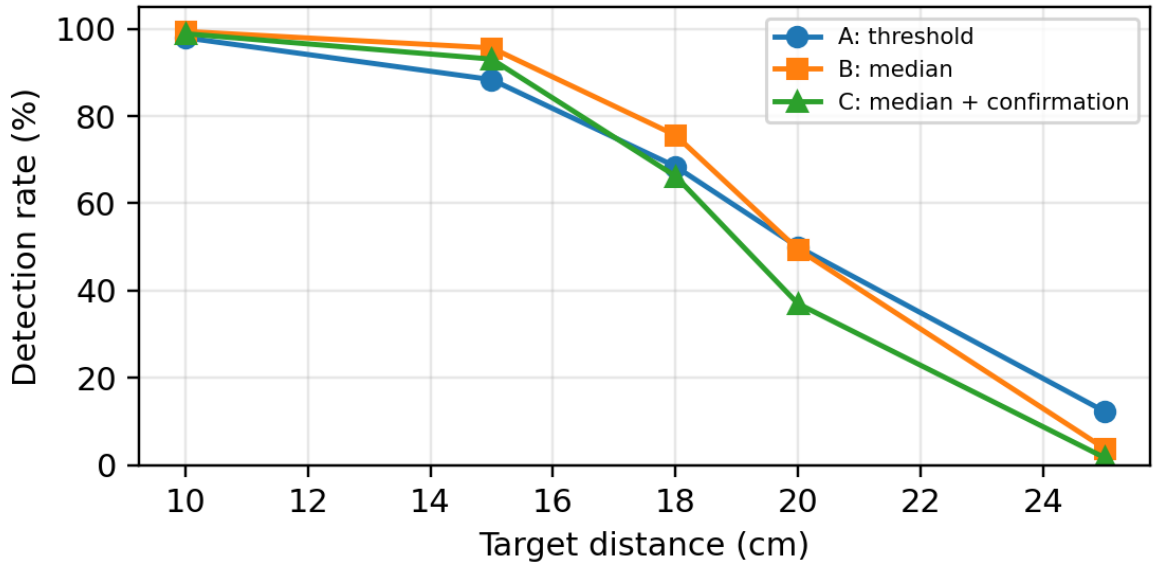


Figure 5. Detection rate versus target distance at σ = 4 cm for Methods A, B and C.

In the 200-trial response-time experiment, all methods achieved 100% detection across the 100-sample event window, but their response delays differed systematically as shown in Figure 6. At σ = 4 cm, mean delay was 0.12 samples for direct thresholding, 1.18 samples for median filtering, and 2.20 samples for median plus confirmation. At σ = 6 cm, the corresponding delays were 0.20, 1.46, and 2.66 samples.

The pre-event trial-level false-alarm percentage tells a complementary story: at σ = 4 cm, 73.5% of baseline trials experienced at least one false alarm prior to event onset, compared with 4.5% for the median method and 1.5% for the confirmation method. At σ = 6 cm, these trial-level rates were 83.0%, 9.5%, and 4.5%, respectively.

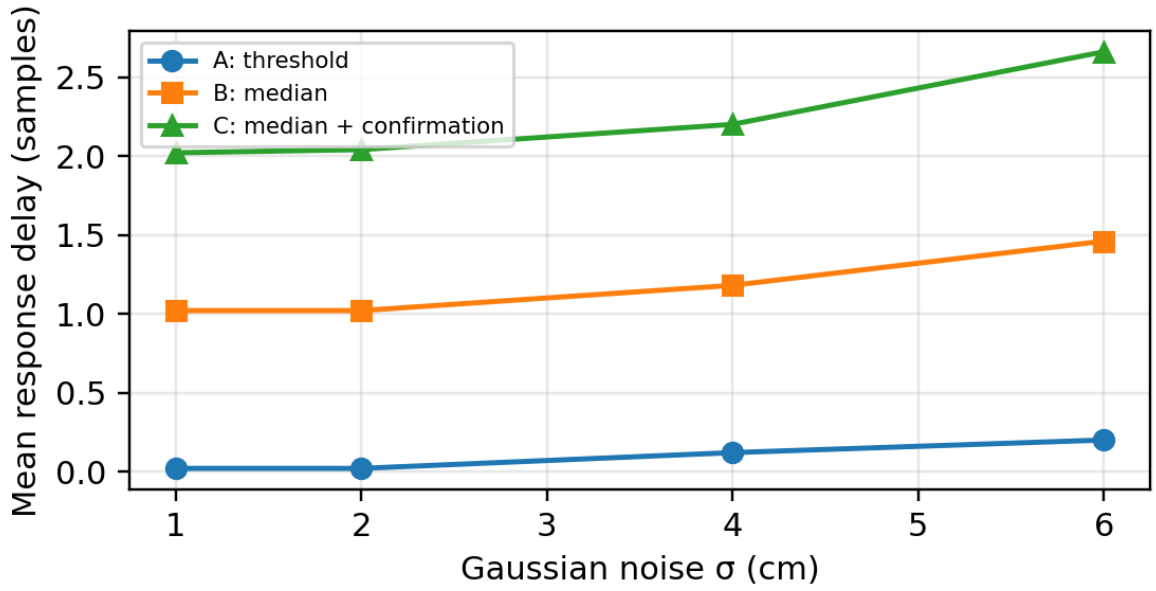


Figure 6. Mean response delay (in simulation samples) versus Gaussian noise standard deviation σ.

4 Discussion

4.1 Practical design implications and parameter sensitivity

Because no single decision rule dominates every metric, system designers should define an acceptable false-alarm budget and maximum response latency prior to deployment. For an embedded monitoring prototype, the three-sample causal median filter offers an effective balance in the simulated setting: it substantially suppresses false alarms while retaining stronger detection at high noise than the confirmation rule. Confirmation is advantageous when nuisance alarms carry a high operational penalty and a delay of two to three samples is acceptable.

Filter-window length and threshold margins also interact strongly with noise. A three-sample median rejects isolated single-sample impulse outliers without excessive lag, whereas longer odd-sized windows suppress multi-sample bursts at the cost of higher transition delay. Near the 20 cm perimeter boundary, symmetric Gaussian noise causes inevitable threshold crossings; incorporating a spatial guard band or entry/exit hysteresis represents a practical refinement for physical deployment. Furthermore, because a servo-mounted sensor samples one angular sector at a time, per-angle sample histories should be maintained rather than sharing a single buffer across distinct scan angles.

4.2 Hardware validation and reproducibility plan

The next stage should validate the algorithm on physical hardware rather than treating synthetic results as field performance. Using the physical Arduino Uno, HC-SR04 sensor, servo, LCD, LEDs, and buzzer, repeated trials should be logged over serial with timestamps, commanded servo angles, raw distances, filtered distances, and binary decisions across three target zones: clearly inside (10–15 cm), boundary-adjacent (18–20 cm), and clearly outside (25–35 cm). Replaying identical recorded raw ultrasonic streams offline across Methods A, B, and C will enable paired statistical comparisons alongside full confusion-matrix reporting and millisecond latency measurement.

Embedded deployment also requires explicit handling of startup buffer initialization—suppressing alarm assertions until three valid samples are acquired—and strict separation of per-angle filter buffers during servo sweeps. For full scientific reproducibility, the source archive should preserve the Arduino firmware sketch, Python simulation scripts, random seeds, raw trial logs, and plotting scripts, keeping the 30-trial robustness dataset, distance sweep dataset, and 200-trial response-time dataset distinct.

5 Conclusion

This paper evaluated direct thresholding, a causal three-sample median filter, and median filtering with two-sample confirmation for a simulated angle-scanned ultrasonic intrusion-monitoring concept. Across 30 independent trials per main noise condition, causal median filtering reduced sample-level false alarms from 1.45–2.07% to 0.31–0.37% while maintaining 87.17% mean detection at $\sigma = 6$ cm compared with 79.32% for direct thresholding. Adding two-consecutive-positive confirmation reduced sample-level false alarms further to 0.26–0.32% and lowered trial-level pre-event false alarms to 1.5–4.5%, while increasing mean response delay to 2.20–2.66 samples. These findings quantify a clear robustness-latency trade-off for embedded ultrasonic sensing and establish a structured baseline for subsequent physical hardware validation.

Declarations

Funding and conflict of interest: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The authors declare that there is no conflict of interest regarding the publication of this manuscript. All listed authors contributed substantially to the simulation design, circuit modeling, analysis, and manuscript preparation, and approved the final version for submission.

Acknowledgment

The authors thank the Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies (RGUKT), RK Valley, Andhra Pradesh, India, for academic guidance and institutional support.

References

[1] Suaif A, Gani FQ. Implementation of Ultrasonic Sensor as a Radar for Obstacle Detection. *Andalas Journal of Electrical and Electronic Engineering Technology*. 2021;1(1):12–15. https://doi.org/10.25077/ajeeet.v1i1.7

[2] Sachkov MY, Yusupova AY. Signal filtration algorithms for ultrasonic sensors in obstacle detection devices. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*. 2019;19(2):326–332. https://doi.org/10.17586/2226-1494-2019-19-2-326-332

[3] Cui H, Li Y, Liu J. An Obstacle Detection Algorithm Based on Ultrasonic Sensors for Autonomous Land Vehicle. In: *Proc*. 2016 International Conference on Mechatronics, Control and Automation Engineering. 2016. p. 147–150. https://doi.org/10.2991/mcae-16.2016.35

[4] Meliones A, Filios C, Llorente J. Reliable Ultrasonic Obstacle Recognition for Outdoor Blind Navigation. *Technologies*. 2022;10(3):54. https://doi.org/10.3390/technologies10030054

[5] Borenstein J, Koren Y. Obstacle avoidance with ultrasonic sensors. *IEEE Journal on Robotics and Automation*. 1988;4(2):213–218. https://doi.org/10.1109/56.2085

[6] Kleeman L, Kuc R. Mobile robot sonar for target localization and classification. *The International Journal of Robotics Research*. 1995;14(4):295–318. https://doi.org/10.1177/027836499501400401

[7] Carullo A, Parvis M. An ultrasonic sensor for distance measurement in automotive applications. *IEEE Sensors Journal*. 2001;1(2):143–147. https://doi.org/10.1109/JSEN.2001.936931

[8] Zhmud VA, Kondratiev NO, Kuznetsov KA, Trubin VG, Dimitrov LV. Application of ultrasonic sensor for measuring distances in robotics. *Journal of Physics: Conference Series*. 2018;1015(3):032189. https://doi.org/10.1088/1742-6596/1015/3/032189

[9] Panda S, Dash SK, Mahapatra RK. Design and implementation of an Arduino-based ultrasonic radar system for real-time object detection and ranging. *International Journal of Engineering Research and Technology*. 2023;12(5):412–416.

[10] Gallagher NC, Wise GL. A theoretical analysis of the properties of median filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981;29(6):1136–1141. https://doi.org/10.1109/TASSP.1981.1163708

[11] Arduino. *UNO R3 documentation* [Internet]. Arduino; 2026 [URL and access date to be supplied by the author].

[12] Atal Tinkering Labs. *Ultrasonic sensor module HC-SR04: ATL equipment manual*. New Delhi: NITI Aayog; 2026.